

## 錬成問題

■ **int** 型の変数  $n1$ ,  $n2$ ,  $n3$  の値が、それぞれ 17, 9, 0 であるとする。このとき、以下の各式を評価した値を示せ。

$n1 \ \& \ n2$	...	(1)	$n2 \ \& \ n3$	...	(2)
$n1 \   \ n2$	...	(3)	$n2 \   \ n3$	...	(4)
$n1 \ \wedge \ n2$	...	(5)	$n2 \ \wedge \ n3$	...	(6)
$n1 \ \gg \ 2$	...	(7)	$n2 \ \ll \ 3$	...	(8)
$n1 \ \& \ (n2 \ \&\& \ n3)$	...	(9)	$n1 \ \& \ (n2 \    \ n3)$	...	(10)

■ **signed** あるいは **unsigned** の (11) 子が与えられていない、“ただの” **int** 型は、(12) であり、“ただの” **char** 型は、(13) である。

▶ (12), (13) の選択肢

- (a) 符号付き型    (b) 符号無し型  
(c) 符号付き型か符号無し型かは処理系によって異なるもの

■ **sizeof(char)** の値は (14) であり、その型は (15) である。

■ 符号無し整数型の演算において、オーバーフローは (16)。

▶ (16) の選択肢 … (a) おこり得る    (b) おこらない

■ **char** 型の表現できる最小値を表す (17)、最大値を表す (18)、**int** 型の表現できる最小値を表す (19)、最大値を表す (20) といった、文字型や整数型の表現範囲を表すためのマクロは <(21)> ヘッド内で定義されている。

■ 配列  $vc$  の要素数・要素型の如何に関わらず、要素数は、(22) / (23) によって得られる。

■ 以下に示すのは、右に示すように、整数  $n1$  から  $n2$  までの値を 8 進、10 進、16 進で表示するプログラム部分である。

なお、各数値は 5 桁の幅で出力するものとする。

```
for (i = n1; (24); i++)
    printf("(25)\n", i, i, i);
```

115	77	4D
116	78	4E
117	79	4F
120	80	50

■ 整数型には、表現できる値の範囲によって、(26)、(27)、(28) の 3 種類があり、右側のものは左側と同じまたはより広い表現範囲をもつ。

■ 浮動小数点型には、(29)、(30)、(31) の三つの型がある。

■ 右に示すのは、符号無し整数 $x$ の全てのビットを反転した値を返す関数である。

```
unsigned revof(unsigned x)
{
    return ( (32) );
}
```

■ 右に示すのは、全てのビットが0である符号無し整数を返す関数である。

```
unsigned all0(void)
{
    return ( (33) );
}
```

■ 右に示すのは、全てのビットが1である符号無し整数を返す関数である。

```
unsigned all1(void)
{
    return ( (34) );
}
```

■ 右に示すのは、 $x$ の最下位ビットを1にした値を返す関数である。

```
unsigned setLSB(unsigned x)
{
    return ( (35) );
}
```

■ 右に示すのは、 $x$ の最下位ビットを0にした値を返す関数である。

```
unsigned resetLSB(unsigned x)
{
    return ( (36) );
}
```

■ 右に示すのは、 $x$ の最下位ビットを反転した値を返す関数である。

```
unsigned invLSB(unsigned x)
{
    return ( (37) );
}
```

■ 右に示すのは、 $x$ の下4ビットを1にした値を返す関数である。

```
unsigned setL4bits(unsigned x)
{
    return ( (38) );
}
```

■ 右に示すのは、 $x$ の下4ビットを0にした値を返す関数である。

```
unsigned resetL4bits(unsigned x)
{
    return ( (39) );
}
```

■ 右に示すのは、 $x$ の下4ビットを反転した値を返す関数である。

```
unsigned invL4bits(unsigned x)
{
    return ( (40) );
}
```

■ 右に示すのは、符号無し整数 $x$ の最下位ビットが0であれば1を、そうでなければ0を返す関数である。

```
int f1(unsigned x)
{
    return ( (41) );
}
```

■ 右に示すのは、符号無し整数 $x$ の下から2ビット目が1であれば1を、そうでなければ0を返す関数である。

```
int f2(unsigned x)
{
    return ( (42) );
}
```

■ 右に示すのは、符号無し整数 $x$ の下から2ビット目が1であれば2を、そうでなければ0を返す関数である。

```
int f3(unsigned x)
{
    return ( (43) );
}
```

■ 右に示すプログラム部分の実行結果を示せ。

```
x + 1 = (44)
x + 10 = (45)
```

```
#include <limits.h>
unsigned x = UINT_MAX;

printf("x + 1 = %u\n", x + 1);
printf("x + 10 = %u\n", x + 10);
```

■ 整数定数には、それが符号無しであることを示す (46) または (47) の、**long**型であることを示す (48) または (49) の整数接尾語を付けることができ、浮動小数点定数には、それが **float** 型であることを示す (50) または (51) の、**long double** 型であることを示す (52) または (53) の浮動小数点接尾語を付けることができる。

■ 右に示すのは、符号無し整数 $x$ を最大 $n$ 桁の2進数で表示する関数である。ただし、 $n$ 桁に満たない場合は、左側に0を埋めて表示する。

```
void print_nbin(unsigned x, int n)
{
    int i;
    for (i = (54); i >= (55); i--)
        putchar((56));
}
```